REMARKS/ARGUMENTS

The Office Action of July 1, 2005 has been carefully reviewed and this response addresses the concerns stated in the Office Action. All objections and rejections are respectfully traversed.

I. STATUS OF THE CLAIMS

Claims 1-35 and 37-64 are currently pending.

Claim 36 was previously canceled without prejudice.

Claims 1-35 and 37-64 are rejected under 35 U.S.C. 102(e) as being anticipated by Tentij et al., U.S. Patent # 6,513,129, issued on January 28, 2003 (Tentij).

Claim 23 is amended to correct an informality. No new matter is added.

II. IMPROPER REJECTIONS

Prior to responding to the rejection of claims 1-35 and 37-64 below, it is respectfully pointed out to the Examiner that the Examiner has presented information from the no longer applied Kekic reference to support the rejection of claims 13, 21, 35, and 46 by Tentij. More specifically, on pages 10 and 11, in paragraph 2, with respect to dependent claims 13, 21, and 46, the Office Action states that Tentij discloses

(1) wherein said at least one distributed polling gateway determining that said one or more user-defined state transition conditions are not satisfied, then the state of said at last one network element remains unchanged (dependent claim 13)

(2) wherein said software code triggers an action upon a user-defined pattern of states of said various different models being achieved (dependent claims 21 and 46) (col. 19, lines 25-58; col. 36, lines 33-67; col. 37, lines 1-4).

Applicant respectfully points out that the cited passages appear to be from the previous Office Action and must refer to the Kekic reference since Tentij does not contain the appropriate number of columns to support this rejection. In particular, column 22 is the last column in Tentij. Therefore, columns 36 and 37 cannot be referring to Tentij. Further, the cited passage, col. 19, lines 25-58, is a cited passage from Kekic. Applicant asserts that no valid reference has been cited against dependent claims 13, 21, and 46.

On pages 6-7, in paragraph 2, with respect to independent claim 35, the Office Action states that Tentij discloses a method for enabling state-based management of a network, wherein network elements are managed based on their state, said method comprising:

(a) receiving input from a user at a management system to define at least one state model for managing at least one network element based on a determined state of said at least one network element (Abstract, col. 5, lines 40-51; col. 6, lines 7-25; col. 15, lines 4-21; col. 19, lines 25-58; col. 24, lines 28-41; col. 36, lines 56-67; col. 37, lines 1-4);

(b) receiving input from a user at said management system to define at least one poll service that includes at leas tone of said at least one state model (Abstract; col. 5, lines 40-51; col. 19, lines 25-58; col. 39, lines 9-26);

(c) distributing said at least one poll service including said at least one state model to at least one distributed polling gateway that is communicatively coupled with said at least one network element;

(d) executing said at least one poll service sat said at least one distributed polling gateway to manage said at least one network element (Col. 19, lines 25-58; col. 77, lines 1-35)

(e) wherein said management system is a central management system (Abstract; col. 5, lines 40-51; col. 19, lines 25-58; col. 39, lines 9-26)

Applicant respectfully points out that the cited passages appear to be from the previous Office Action and must refer to the Kekic reference since Tentij does not contain the appropriate number of columns to support this rejection. In particular, column 22 is the last column in Tentij. Therefore, columns 24, 36, 37, 39, and 77 cannot be referring to Tentij. Further, there is no cited reference against clause (c) above. Applicant asserts that no valid reference has been cited against independent claim 35.

III. REJECTION OF CLAIMS 1-35 AND 37-64 UNDER 35 USC § 102(e) AS BEING ANTICIPATED BY TENTIJ

Please note that Applicant presents the following arguments, with a few exceptions, in claim order.

Applicant respectfully points out that Tentij issued on January 28, 2003, almost two years after the filing date of the present application, January 26, 2001. Applicant is investigating the possibility of swearing behind the cited reference and respectfully reserves the right to file a petition under 37 C.F.R. § 1.131.

Applicant further respectfully points out that "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628 (CAFC, 1987), M.P.E.P. § 2131. As provided by the remarks set forth below, clearly this is not the case with the present rejection of the claims. For example, although there are numerous other claimed limitations lacking as pointed out below, the limitations presented below in the independent claims as well as the dependent claims are provided as a basis for clearly setting forth the lack of Applicant's claimed subject matter in Tentij.

(1) Nowhere does Tentij disclose or suggest Applicant's claimed at least one poll service that includes at least one state model (independent claims 1 and 35); and

(2) Nowhere does Tentij disclose or suggest Applicant's claimed executing at least one poll service at said at least one distributed polling gateway to manage said at least one network element (independent claims 1, 35, and 59); and

(3) Nowhere does Tentij disclose or suggest at least one state model executing on said one or more distributed gateways for managing said at least one network element based on a determined state of said at least one network element, said at least one state model capable of being dynamically defined during runtime (independent claims 48 and 64); and

(4) Nowhere does Tentij disclose or suggest executing, on at least one distributed gateway located between the central management system and the network elements, at least one user-defined state model for managing at least one network element based on a determined state of said at least one network element, wherein said executing at least one user-defined state model includes polling said at least one network element for data, evaluating said data to determine whether a user-defined state transition condition is satisfied, and triggering a state transition if said user-defined state transition condition is satisfied for a user-defined number of consecutive polls of said at least one network element (claim 59).

On pages 5-6 of Applicant's specification, Applicant's claimed polling services are discussed with respect to communication between gateways and network elements. It is these polling services that are augmented in the present invention. No such polling services are disclosed or suggested by Tentij, who, on the contrary, discloses a system that awaits a fault notification from a network element before initiating partial processing in a gateway (col. 5, lines 5-9).

Further, and more importantly, although a two-way communication is shown by Tentij in FIG. 5A between gateway 225 and network elements 210, in fact, the disclosed system, for example as described by Tentij in FIG. 7, indicates a one-way communication between gateway 225 and network elements 210. This one-way communication teaches away from Applicant's claimed gateway managing at least one network element (claims 35, 48, 59, and 64).

Note that dependent claims 2-34 depend upon independent claim 1, dependent claims 37-47 depend upon independent claim 35, dependent claims 49-58 depend upon independent claim 48, and dependent claims 60-63 depend upon independent claim 59.

On pages 2, 7, 9, and 11-12, in paragraph 2, with respect to independent claim 1 and dependent claims 2, 18, 43, and 50, the Office Action states that Tentij discloses

(1) a method for implementing a state model for managing at least one distributed network element communicatively coupled to a central management system, said method comprising: presenting a user interface for said central management system to enable a

user to define at least one poll service that includes at least one of said at least one state model (independent claim 1);

(2) communicatively coupling said user interface to said at least one distributed polling gateway (dependent claim 2);

(3) wherein said presenting a user interface on a management system to enable a user to define at least one state model further comprises providing a user interface that allows a user to define a plurality of states within a state model for a network element, and providing a user interface that allows a user to define at least one transition condition that specifies when a transition from one of said plurality of states to another of said plurality of states is to occur (dependent claims 18 and 43);

at least one user-defined poll service that includes one or more of said at least one state model (dependent claim 50) (col. 5, lines 8-22; col. 6, lines 6-20).

Applicant asserts that Tentij does not disclose a user interface that enables a user to define a poll service that includes at least one state model (claims 1 and 50). Nor does Tentij disclose a user interface that allows a user to define a plurality of states within a state model for a network element, or a user interface that allows a user to define a transition condition that specifies when a transition from one state to another occurs (claims 18 and 43). In the cited passages, Tentij states that object processing in a gateway and a management processor system carries out the predefined policies as defined by a user, and a user interface provides users with interactive access to a management system through the gateway or the management processor system. In other words, Tentij states that a user can define policies, but even a broad reading of Tentij cannot produce an interpretation that would anticipate Applicant's claimed user interface for defining a poll service. Further, Tentij could not inherently disclose a user interface for defining a poll service because nowhere does Tentij disclose any form of polling.

On pages 3 and 12, in paragraph 2, with respect to independent claim 1 and dependent claim 51, the Office Action states that Tentij discloses

(1) a method for implementing a state model for managing a network, said method comprising presenting a user interface for said central management system to enable a user to define at least one poll service that includes at least one of said at last one state model (independent claim 1) (col. 1, lines 40-48; col. 5, lines 8-22; col. 6, lines 16-20);

(2) software executing on said central management system to enable a user to define said at least one poll service, wherein once a user defines said at least one poll service, it is communicated to said one or more distributed gateways for execution thereon (dependent claim 51) (col. 1, lines 40-48; col. 5, lines 8-22).

Applicant asserts that Tentij does not disclose a user interface to enable a user to define a poll service (claim 1), nor software to enable a user to define a poll service (claim 51). In the cited passages, Tentij states that

(a) a gateway is connected to a network;

(b) the gateway receives alarm incidents from the network;

(c) the gateway has a rule engine to select and process a control object;

(d) the gateway selects and partially processes (basic processing) a control object in response to a received incident;

(e) the remainder of the processing (advanced processing) is done in a management processor system;

(f) the management processor system processes configuration objects in response to the selected control object;

(g) the processing done by the gateway and management processor system implements fault management objectives defined by the user; and

(h) a user interface provides users with interactive access to the management system through the gateway or through the management processor system.

In other words, Tentij states that a gateway and a management processor implement objectives that are defined by a user. Nowhere in this very detailed list does Tentij disclose Applicant's claimed user interface or software to define a poll service, and delivery of the poll service to one or more distributed gateways. To reiterate, the poll service as defined in Applicant's specification on pages 5-6 is nowhere included in the disclosure of Tentij, and therefore Tentij could not possibly include a user interface or software to define a poll service. A poll service as defined by Applicant, polls network elements to determine whether one or more user-defined state transition conditions are satisfied (Applicant's specification, page 15, lines 7-8). Whereas Tentij receives alarms from network elements and decides how to process the alarm based on a match between the name of the alarm and a list of names of control objects, Applicant claims a poll

service that is defined to poll the network elements and determine if state transitions have occurred.

On page 3, in paragraph 2, with respect to independent claim 1, the Office Action states that Tentij discloses a method for implementing a state model for managing a network, said method comprising executing said at least one poll service to manage said at least one network element (col. 8, lines 10-26).

Applicant asserts that Tentij does not have an equivalent concept to Applicant's poll service. In the cited passage, Tentij states that the gateway can include distributed gateways. Also, Tentij states that the processing flow of an incoming alarm incident proceeds as follows: (a) a network element sends an alarm incident to a gateway, (b) the alarm incident is identified and translated from a vendor-specific form into a form that can be processed in Tentij's system, and (c) the gateway selects the control object that has a name that most closely matches the alarm incident name. In other words, the network element initiates action in the system of Tentij, whereas in Applicant's system, on the contrary, the poll service (defined previously) initiates action and manages the network element. In the system of Tentij alarm incidents are managed through user-defined fault management objectives (col. 1, lines 45-47), while in the system of Applicant, network elements are managed by a poll service (claim 1), where the poll service is defined in Applicant's specification (p. 15, line 7) to track state transitions.

On pages 6, 7, and 9-10, in paragraph 2, with respect to dependent claim 2 and independent claims 48 and 64, the Office Action states that Tentij discloses

(1) distributing said at least one poll service to at least one distributed polling gateway that is communicatively coupled with said at least one network element (dependent claim 2);

(2) a system for managing network elements based on their state, said system comprising one or more distributed gateways for monitoring said at least one network element, said one or more distributed gateways communicatively coupled to a central management system between said at least one network element and said central management system (independent claim 48);

(3) at least one gateway for monitoring said at least one network element, said at least one gateway communicatively coupled to a central management system between said

at least one network element and said central management system (independent claim 64) (col. 5, lines 8-22).

Applicant asserts that Tentij does not disclose or suggest Applicant's claimed step of distributing a poll service to a distributed polling gateway. Tentij states that multiple incident-receiving gateways may be implemented with a centralized management processor system. However, nowhere does Tentij disclose or suggest a poll service that is distributed to a polling gateway. The gateways of Tentij can receive alarms, but, as stated previously, no poll service is distributed to them. Further, nowhere does Tentij disclose or suggest Applicant's claimed managing network elements based on their state. As stated previously, Tentij states that alerts are received from network elements into gateways, but nowhere does Tentij disclose communication in the reverse direction. Therefore, Tentij cannot anticipate Applicant's claimed managing network elements based on their state.

On pages 7, 8, and 11, in paragraph, with respect to dependent claims 3, 4, 16, and 37, the Office Action states that Tentij discloses distributing said at least one poll service defined by said user (dependent claims 3 and 37), distributing said at least one poll service defined by said user a plurality of distributed polling gateways for execution thereon (dependent claim 4), and wherein said at least one distributed polling gateway determining that said one or more user-defined state transition conditions are satisfied in a user-defined number of consecutive polls of said at least one network element, than a state transition for said at least one network element is triggered (dependent claim 16) (col. 1, lines 40-48; col. 8, lines 10-26).

The shortcomings with respect to the cited passages have been set forth and will not be repeated here. As previously stated, Applicant asserts that Tentij does not disclose or suggest a poll service, and therefore could not anticipate Applicant's claimed poll service (claims 3, 4 and 37), nor can Tentij anticipate Applicant's claimed user-defined number of consecutive polls (claim 16). In particular, Applicant's claimed user-defined number of consecutive polls is used to indicate how many times a network element is polled to determine if a state transition condition is satisfied, thus determining when the transition is actually triggered (Applicant's specification, page 15, lines 11-13). Applicant fails to understand how the cited passage or the reference as a whole anticipates Applicant's dependent claims 3, 4, 16, and 37.

On page 8, in paragraph 2, with respect to dependent claims 6 and 38, the Office Action states that Tentij discloses wherein said at least one distributed polling gateway filters data (col. 5, lines 5-15).

In the cited passage, Tentij states that a gateway selects a control object in response to a received incident, and that basic processing of the control object is accomplished in the gateway. Tentij states that multiple incident-receiving gateways may be implemented with a centralized management processor system. In other words, Tentij uses the incident as an index to determine which control object to retrieve and process, i.e. Tentij is processing the control object, not the incident, which is the incoming data. Applicant, on the contrary, claims a distributed polling gateway that filters data for a central management system.

On page 10, in paragraph 2, with respect to dependent claims 7, 12, 14, 39, 40, 41, 52, 53, and 54, the Office Action states that Tentij discloses wherein said at least one distributed polling gateway communicating data satisfying said at least one state model to said central management system (claims 7, 39, and 52), wherein said at least one distributed polling gateway executing software to evaluate one or more user-defined state transition conditions for said at least one state model to determine whether said one or more user-defined state transition conditions are satisfied (claims 12, 40, and 53), and wherein said at least one distributed polling gateway determining that said one or more user-defined state transition conditions are satisfied, then a state transition for said at least one network element is triggered (claims 14, 41, and 54) (col. 5, lines 8-22; col. 8, lines 1-10).

In the first cited passage, Tentij states that a selected control object is processed in a gateway and a management processor system. In the second cited passage, Tentij states that a display terminal displays user fault alerts and network topological information, receives command and configuration information from the user, and is connected to the gateway and the management processor system. Tentij also states that a gateway is interconnected between a management processor system and network elements and receives fault incidents from the network elements and element managers. In other words, the gateway of Tentij receives alerts, partly processes them and then sends them on to the

management processor system for further processing. Tentij states that "basic" processing is done by the gateway and "advanced" processing is done by the management processor system. Thus all that are needed for advanced processing are transferred from the gateway to the management processor system, regardless of the state of the gateway, the management processor system, the control object, or the incoming alert. Applicant, on the contrary, claims a distributed polling gateway that communicates data satisfying at least one state model. Applicant's gateway polls network elements, may receive data from those network elements, and communicates those data that satisfy at least one state model. Tentij discloses no such restriction on data transmission.

Further, nowhere does Tentij disclose state transitions or a state model. Tentij states that a rule engine in the management processor system performs advanced processing of selected control objects that have been first processed by the gateway. Applicant, on the contrary, claims a polling gateway that, if a state transition condition is satisfied, (a situation determined through polling by the gateway of a network element) a state transition for the network element is triggered. A rule engine does not necessarily detect state transition conditions and trigger state transitions as a result, and nowhere does Tentij disclose or suggest that the disclosed rule engine performs that function. Further, even if the rule engine could be construed to be testing for state transition conditions, the rule engine is performing these functions is executing in the management processor system, not the gateway, of Tentij. The "rule engine" in the gateway of Tentij simply identifies the incoming alert in order to determine an associated control object. Nowhere does Tentij disclose or suggest Applicant's claimed invention as set forth in dependent claims 7, 12, 14, 39, 40, 41, 52, 53, and 54.

On pages 3, 8, 9, 11, in paragraph 2, with respect to dependent claims 8, 15, 17, 18, 25, 27, 42, 43, 49, 55, and 61, the Office Action states that Tentij discloses

(1) wherein said at least one distributed polling gateway executing software to evaluate a user-defined state model condition to determine whether to execute each of said at least one state model (dependent claim 8);

(2) wherein one or more user-defined transition actions for said state transition are triggered in response to said state transition (dependent claims 15, 17, 42, and 55);

(3) providing a user interface that allows a user to define at least one transition action to be performed upon the occurrence of said transition (dependent claims 18 and 43);

(4) wherein said at least one state model includes software code specifying at least two user-defined states for said at least one network element, software code specifying at least one transition from a first of said at least two user defined states to a second of said at least two user-defined states, and software code specifying at least one transition action to be performed upon the occurrence of said at least one transition (dependent claim 25);

(5) wherein said transition action includes any one or more selected from the group consisting of enabling a particular poll service for said at least one network element, disabling said particular poll service for said at least one network element, enabling a particular state model for said at least one network element, disabling said particular state model for said at least one network element, and triggering one or more user-defined commands to be executed (dependent claim 27); and

(6) wherein said at least one distributed polling gateway software executing on said central management system to enable a user to define said at least one state model, wherein once a user defines said at least one state model (dependent claims 49 and 61) (col. 8, lines 1-10)

In the cited passage, Tentij states that a display terminal, connected to both a gateway and a management processor system, can display fault alerts received by the gateway. Tentij states that the display terminal can receive user command and configuration information, topological representations of the network, and its state. Tentij also states that the gateway is interconnected between the management processor system and network elements. In other words, the cited passage lays out the elements and connectivity of the fault management system of Tentij, and states two aspects of data flow: (a) fault alerts are received by the gateway, and (b) command and configuration

information, network topology, and network state are received by the display terminal. According to the passage, the display terminal is connected to both the gateway and the management processor system. Elsewhere in the disclosure, Tentij states that advanced processing is accomplished in the management processor system (not the gateway). Further, network topology and network state are commonly understood to mean snapshot views of the current status of the network.

Nowhere in the cited passage does Tentij state what the gateway is doing beyond receiving and identifying alerts and performing basic processing on them. Further, it cannot be inferred from the passage's reference to "command and configuration information" that the gateway is executing software to evaluate a user-define state model condition (claim 8) because nowhere in the cited passage or in the disclosure is there a reference to such an action by the gateway. The cited passage does not allude to a state model or state transitions at all, neither of which is equivalent to the current status of the network (dependent claims 15, 17, 42, and 55). It cannot be inferred that the command and configuration information, network topology, and network state that can be entered and viewed at the display terminal of Tentij includes Applicant's claimed transition action to be performed upon the occurrence of said transition (dependent claims 18, 25, 27, 43, 49, and 61) because the disclosure does not support the use of such information in any way. There is no reference in the disclosure to state transitions.

On page 8, in paragraph 2, with respect to dependent claim 9, the Office Action states that Tentij discloses wherein said state model condition specifies that said at least one state model is to be executed only for particular network elements (col. 8, lines 1-10; col. 9, lines 62-66).

The shortcomings of the first cited passage have been set forth previously and will not be repeated here. In the second cited passage, Tentij states that the gateway parses and identifies incoming alarm incidents. Tentij states that control objects are used by the gateway to find the correct behavior specification for the alarm incidents. In other words, Tentij identifies a control object to process an alarm according to the type (name) of the alarm. On the contrary, Applicant claims a state model condition that specifies a state model that is to be executed for particular network elements, not for a particular alarm.

Nowhere does Tentij disclose or suggest a situation in which a state model is selected and executed for particular network elements.

On pages 9, 11, and 12, in paragraph 2, with respect to dependent claims 19, 20, 44, 45, and 56, the Office Action states that Tentij discloses

(1) correlating various different models of said at least one state model (dependent claims 19 and 44);

(2) wherein software code executes on at least one distributed polling gateway communicatively coupled to said central management system to perform said step of correlating (dependent claims 20 and 45); and

(3) at least one pattern-based state model executing thereon to correlate various of said at least one state model (dependent claim 56) (col. 12, lines 49-55).

In the cited passage, Tentij states a condition under which a correlation alert is displayed and regenerated, for example, if alarms are paired. Tentij also states that advanced processing, which occurs on the management processing system, includes correlation (col. 11, lines 35-37). In other words, Tentij correlates alerts with one another in the management processing system, not the gateway. Applicant, on the contrary, claims correlating *state models* on a gateway. Thus, even if correlating alerts could be construed to broadly include the correlation of state models, a premise with which Applicant does not agree, Tentij does not perform correlation on a gateway, as Applicant has claimed.

On pages 12-13, in paragraph 2, with respect to dependent claims 22, 26, 47, 58, and 63, the Office Action states that Tentij discloses wherein said action includes any one or more selected from the group consisting of generating a user alert, clearing said user alert, starting particular services for said at least one network element, stopping particular services for said at least one network element, changing the interval utilized to poll said at least one network element, enabling a particular poll service for said at last one network element, disabling aid particular poll service for said at least one network element, enabling a particular state model for said at least one network element, disabling said particular state model for said at least one network element, triggering one or more user-

defined commands to be executed, triggering communication of an email message to personnel, triggering a page of personnel, logging achievement of said pattern of states to a file, and performing network element configuration (col. 8, lines 52-67).

In the cited passage, Tentij states a naming convention for control objects. The cited passage does not, however, disclose or suggest Applicant's claimed action upon a user-defined pattern of states selected from a group consisting of the above-listed activities.

On page 4, in paragraph 2, with respect to dependent claims 28 and 30, the Office Action states that Tentij discloses

(1) wherein said executing said at least one poll service further includes triggering. execution of said at least one poll service in response to the occurrence of a user defined event (dependent claim 28); and

(2) wherein said at least one poll service is executed only if a user-defined activation condition for said at least one poll service is satisfied (dependent claim 30) (col. 1, lines 40-48; col. 8, lines 1-10).

The shortcomings of the cited passages have been set forth above with respect to claims 1 and 7 and will not be repeated here. Nowhere in either passage does Tentij disclose or suggest a poll service. A poll service as defined by Applicant involves a gateway's polling a network element, which implies a communication path from the gateway to the network element. Tentij teaches away from such a communication path because Tentij states that the gateway is communicatively connected to a network for receiving alarm incidents from the network, implying a one-way communication path, an implication that is substantiated throughout Tentij's disclosure.

On page 4, in paragraph 2, with respect to dependent claim 31, the Office Action states that Tentij discloses wherein said user-defined activation condition specifies that said poll service is for a particular type of network element (col. 9, lines 62-66).

The shortcomings of the cited passage have been set forth previously and will not be repeated here. As Applicant has previously stated, nowhere does Tentij disclose or suggest a poll service. Tentij states that control objects are assigned to particular incoming alarm incidents (according to name). Applicant claims that a poll service is for a particular type of network element. Thus, even if Tentij could be construed to anticipate Applicant's poll service with Tentij's control object, a premise with which Applicant does not agree, Tentij draws a relationship between control objects and incoming alarm incidents, whereas Applicant draws a relationship between a poll service and a network element. A network element could produce an alarm, but a network element is not equivalent to an alarm. For example, an alarm such as "timeout" could be received from many types of network elements. In the system of Tentij, all such alarms are related to a certain control object. Conversely, a network element could produce many types of alarms such as "out of paper" or "online". In the system of Tentij, each of these would be processed by a different control object, whereas in the system of Applicant each of these would be processed by Applicant's claimed poll service that is specified for a particular type of network element.

On page 4, in paragraph 2, with respect to dependent claim 32 and independent claim 64, the Office Action states that Tentij discloses

(1) wherein said central management system enables a user to dynamically define said at least one poll service during runtime (dependent claim 32);

(2) at least one state model executing on said at least one gateway for managing said at least one network element based on a determined state of said at least one network element, said at least one state model capable of being dynamically defined during runtime (independent claim 64) (col. 5, lines 8-22; col. 11, lines 52-57).

In the first cited passage, Tentij states that the management processor system may be implemented with a centralized server and that multiple gateways may be implemented with a centralized management processor system. Tentij also states that one of the advanced processing attributes can contain the name of an event that runs before system default functionality is triggered. Tentij states that this event can be used to pre-set

attributes that the system uses during run time. In other words, if one of the attributes that is processed during advanced processing by the management processor system contains an event name, then the event by that name runs before any system defaults are set, and attributes that are used during run time are set. The description implies a sequential activity with a static result, i.e. the setting of attributes that are later used. On the contrary, Applicant claims that the user can dynamically define at least one poll service during runtime. This means that, while the system is executing, not pre-run time as in the system of Tentij, the user can define a poll service. Nowhere does Tentij disclose a poll service, nor dynamic definition of a poll service.

On pages 4, 5 and 10, in paragraph 2, with respect to dependent claims 33 and 34, and independent 48, the Office Action states that Tentij discloses

(1) wherein said central management system enables a user to dynamically define said at least one state model during runtime (dependent claim 33);

(2) wherein said central management system enables a user to dynamically modify an existing poll service or state model during runtime (dependent claim 34); and

(3) a system for managing network elements based on their state, said system comprising at least one state model and managing said at least one network element based on a determined state of said at least one network element, said at least one state model capable of being dynamically defined during runtime (dependent claim 48) (col. 8, lines 1-10; col. 11, lines 52-57).

The shortcomings of the first cited passage have been set forth with respect to claim 8 and will be expanded upon here. The shortcomings of the second cited passage have been set forth with respect to claim 32 and will also be expanded upon here. Tentij states that a display terminal can receive user commands and configuration information and can display topological representations of the network and its state. Tentij states the semantics of a pre-process attribute that is accessed during advanced processing of an alert and can provide an event to be executed to provide attributes for processing the alert.

In other words, the display terminal can display information about the network and can allow input of user commands. There is no connection stated or implied between the user commands and any communication between a gateway and a network element. Further, although a network element could be providing information about the network to the gateway to be displayed in the form of topological or status information, there is no communication stated or implied from the gateway to the network element. On the contrary, Applicant claims a state model executing on a gateway to manage a network element, a configuration that requires communication from the gateway to the network element. Further, Tentij states that the pre-process attribute allows for setting attributes in advance of the system's default functionality being triggered. This indicates sequential processing, that the attributes are set, and then execution is begun. On the contrary, Applicant claims a state model that can be dynamically defined at runtime. Dynamic definition is commonly known is the art as definition that occurs anytime the system is executing, not in any particular order.

In neither passage does Tentij disclose or suggest Applicant's claimed state model and its dynamic definition, nor Applicant's claimed distributed gateway for managing a network element.

On page 12, in paragraph 2, with respect to dependent claim 57, the Office Action states that Tentij discloses wherein said at least one pattern-based state model specifies a user-defined pattern of states of said various models, and wherein said at least one pattern-based state model triggers an action upon said user-defined pattern of states being achieved (col. 10, lines 57-60; col. 12, lines 49-55). In the cited passages, Tentij states that actions are triggered based on received alerts or correlated alerts. Applicant, on the contrary, claims a user-defined pattern of states of various models that trigger an action. Whereas Tentij's alerts are correlated, Applicant claims correlated state models (dependent claim 56) that trigger actions (dependent claim 57).

On pages 5 and 6, in paragraph 2, with respect to in dependent claim 59 and dependent claims 60 and 62, the Office Action states that Tentij discloses

(1) a method for performing state-based management of a network, wherein network elements are managed based on their state, said method comprising executing, on at least one distributed gateway located between the central management system and the network elements at least one user-defined state model for managing at least one network element based on a determined state of said at least one network element, wherein said executing at least one user-defined state model includes polling said at least one network element for data, evaluating said data to determine whether a user-defined state transition condition is satisfied, and triggering a state transition if said user-defined state transition condition is satisfied for a user-defined number of consecutive polls of said at least one network element (independent claim 59) (col. 1, lines 40-48; col. 5, lines 8-22; col. 8, lines 1-10);

(2) wherein said user-defined number of consecutive polls is a plurality of polls (claim 60) (col. 1, lines 40-48);

(3) wherein if said user-defined state transition condition is satisfied for a user-defined number of consecutive polls of said at least one network element, then one or more user-defined transition actions for the user defined state transition are triggered (dependent claim 62) (col. 1, lines 40-48; col. 5, lines 8-22; col. 10, lines 57-62).

The shortcomings of the cited passages have been set forth above and will not be repeated here. As previously stated, nowhere does Tentij disclose or suggest Applicant's claimed polling said at least one network element for data, and therefore, nowhere does Tentij disclose or suggest a user-defined number of consecutive polls of at least one network element. The rationale for Applicant's assertion can be found in previous arguments.

Applicant asserts that dependent claims 5, 10, 11, 24, and 29 even more specifically define the invention and should therefore be allowed for the same reasons as the claims from which they depend.

Since Tentij does not anticipate each and every element of Applicant's independent claims 1, 35, 48, 59, and 64, either expressly or inherently, Applicant's

independent claims 1, 35, 48, 59, and 64 (as well as claims 2-34, 37-47, 48-58, and 60-63 that depend, either directly or indirectly, therefrom and that further define the invention) are not anticipated by Tentij, and a rejection under 35 U.S.C. § 102(e) is inappropriate. Applicant asserts that independent claims 1, 35, 48, 59, and 64 (as well as claims 2-34, 37-47, 48-58, and 60-63 that depend, either directly or indirectly, therefrom) are now in condition for allowance. Applicant respectfully requests the withdrawal of rejections under 35 U.S.C. § 102(e) for the reasons set forth above. Furthermore, a 35 U.S.C. § 103 rejection of these claims would be inappropriate as well. Applicant's claimed invention is not an obvious extension of the use of Tentij to meet Applicant's patentable limitations.

IV. CONCLUSION

In view of the absence from any cited reference of Applicant's claimed invention as set forth above, Applicant respectfully urges that Tentij is not sufficient to render the presently claimed invention anticipated under 35 U.S.C. 102(e). Further, Applicant asserts that since the comments presented by the Examiner with respect to claims 13, 21, 35, and 46 were not taken out of Tentij, the applicability of the 35 U.S.C. § 102(e) rejections are clearly without merit. Applicant therefore asserts that independent claim 35 and the dependent claims 13, 21, and 35-47 are in condition for allowance.

Since independent claims 1, 35, 48, 59, and 64 are believed to be in condition for allowance for the reasons stated above, all dependent claims are believed to be in condition for allowance as well by virtue of the more limiting nature of their claim language and their dependence upon allowable independent claims.

No fees are anticipated. However, the Commissioner for Patents is authorized to charge any further additional fees or credit overpayment to Deposit Account No. 50-1078.

The following information is presented in the event that a call may be deemed desirable by the Examiner:

JACOB N. ERLICH (617) 854-4000

Respectfully submitted,
Semih Secer, Applicant

Date: September 30, 2005

By: 
Jacob N. Erlich
Reg. No. 24,338
Attorney for Applicant